7. Game of Life

Program Name: GameOfLife.java Input File: gameoflife.dat

Conway's game of life tries to show how the functions of life -- birth, growth, reproduction and death -- can be mimicked by very simple rules. For this program you will be given a 2-D grid with 1's (representing living cells) and 0 (representing dead cells). This 2-D world evolves in time as per simple rules. The rules are the following:

1. If a cell is immediately surrounded by more than 3 creatures (counting up its neighbors in the 8 directions left-right, up-down, and diagonally), it dies by overcrowding.
2. If a cell has two or three neighbors, it lives for the next round.
3. If a cell has less than two neighbors, it dies due to lack of companionship.
4. A dead cell with exactly three neighbors comes alive in the next round, as if by reproduction.

The initial pattern constitutes the *seed* of the system. The first generation is created by applying the above rules simultaneously to every cell in the seed—births and deaths occur simultaneously, and the discrete moment at which this happens is sometimes called a *tick* (in other words, each generation is a pure function of the preceding one). The rules continue to be applied repeatedly to create further generations.

Input

The first N+2 lines have the data for each test case.
The first line is N, the size of the square grid.
The next line is M, the number of generations your program has to let the grid world evolve.
The next N lines are 1's and 0's representing each row's cells.

Output

Print the grid world as it would look after M generations. You may assume that any cells beyond the boundary are dead. A blank line separates multiple outputs.

Example Input

```
2
5
3
00110
01010
01110
00110
00000
10
5
0000111110
1010000000
1001011110
1111001100
0000001100
0000111000
0000011000
0000011100
0000000110
0000001110
```

Example Output

```
00110
11001
11011
01110
00000

0000000000
0000000000
1010000000
0010000000
1010000000
0000000000
0000000000
0000001000
0000001000
0000001000
```